

The NORMA Software Tool for ORM 2

Matthew Curland¹ and Terry Halpin²

¹ LogicBlox, USA

² LogicBlox, Australia and INTI International University, Malaysia
{matthew.curland, terry.halpin}@logicblox.com

Abstract. Second generation Object-Role Modeling (ORM 2) is a prime exemplar of fact-orientation, an approach that models the underlying facts of interest in an attribute-free way, using natural sentences to identify objects and the roles they play in relationships. ORM 2 provides languages and procedures for modeling and querying information systems at a conceptual level as well as mapping procedures for transforming between ORM structures and other structures, such as Entity Relationship (ER) models, class models in the Unified Modeling Language (UML), relational database models, extensible markup language schemas (XSD), and datalog. This paper provides an overview of Natural ORM Architect (NORMA), an ORM 2 tool under development that is implemented as a plug-in to Microsoft Visual Studio. For data modeling purposes, ORM typically provides greater expressive power and semantic stability than provided by UML or industrial versions of ER. NORMA's support for automated verbalization and sample populations facilitates validation with subject matter experts, and its live error-checking provides efficient feedback to modelers.

1 Introduction

Fact-oriented modeling is a conceptual approach (including languages and procedures) for modeling, transforming, and querying information, that specifies the fact structures of interest as well as the applicable business rules in terms of concepts that are intelligible to the business users. Unlike Entity-relationship modeling (ER) [4] and class diagramming in the Unified Modeling Language (UML) [18], fact-orientation makes no use of attributes as a way to express facts, instead representing all ground assertions of interest as atomic (non-decomposable) facts that are either existential facts or elementary facts. An existential fact simply asserts the existence of an entity (e.g. There is a country named 'Australia'). An elementary fact predicates over individuals (objects that are either entities or values). For example, "The Country named 'Australia' is large" expresses a unary fact about an entity, and "The Country named 'Australia' has the Nickname 'Down Under'" expresses a binary fact that relates an entity to a value.

Elementary facts are expressed using mixfix predicates, and are instances of *fact types*. For example, the UML attributes `Person.isSmoker` and `Person.birthdate` are modeled instead as `Person smokes` (unary fact type) and `Person was born on Date` (binary fact type). Higher arity fact types are allowed, for example `Person played Sport for Country` (a ternary)